UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/332,263	06/11/1999	MICHAEL C. ADLER	5231.8-4015	2707

38492 7590 01/20/2004

WILKIE FARR & GALLAGHER LLP
INTELLECTUAL PROPERTY LEGAL ASSISTANTS
787 SEVENTY AVE
NEW YORK, NY 10019-6099

EXAMINER	
TANG, KENNETH	

ART UNIT	PAPER NUMBER
2127	16

DATE MAILED: 01/20/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/332,263	ADLER ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 October 2003*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22 and 24-38* is/are rejected.

7)☒ Claim(s) *23* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *11/3/00* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *14*.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This final action is in response to Paper Number 13, Response/Amendment A, received on 10/22/03. Applicant's arguments have been fully considered but they are not deemed to be persuasive.

2. Claims 1-38 are presented for examination.

Specification

3. The references for IDS of copending application, submitted in Paper #3 (11/3/00), needs to be stated within the Cross-References to Related Application section of the specification.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-6, 10-11, 14, 17, 19-20, 22, 24-25, and 27-31, and 33-37 are rejected under 35 U.S.C. 102(e) as being unpatentable over Chernoff et al. (hereinafter Chernoff) (US 6,000,028).

4. As to claim 1, Chernoff teaches the invention as claimed including a computer, comprising:

- an instruction pipeline configured to execute instructions of the computer *("pipeline instructions and thus execute the application program more quickly and efficiently", col. 20, lines 9-11, "pipelined fashion, executing multiple instructions in parallel", col. 19, lines 45-46)*;
- profile circuitry configured to detect and record, without compiler assistance for execution profiling, profile information describing a sequence of events occurring in the instruction pipeline, the sequence including every event occurring during a profiled execution interval that matches time-independent selection criteria of events to be profiled, the recording continuing until a predetermined stop condition is reached, the profile circuitry further configured to detect the occurrence of a predetermined condition, after a non-profiled interval of execution, and to thereon commence of the profiled execution interval *(see Figure 1, "condition codes are maintained on an X86 machine to provide knowledge of the machine state", col. 2, lines 64-66, "query the condition code", "conditional branch", col. 3, lines 4-7, "storing, in response to an execution of a condition code modifying instruction, a first table of methods, the methods providing access to evaluation routines which, when executed, modify one or all of the plurality of condition code bits", col. 3, lines 50-61, "a profile file data structure 60 used to store information, col. 13, lines 62-67, see Fig. 5, "fields 70 are used to keep track or maintain count of the targets of the control transfer instruction", " number of times a conditional*

control transfer of a branch instruction was taken", col. 14, lines 64-67 through col. 15, lines 1-3).

It is inherent that Chernoff teaches the profiling to be done without compiler assistance because Chernoff does not disclose using compiler assistance. The various condition codes represent the criteria of events of various machine states, which are predetermined. A specific example, if and only if a condition is met that would allow for information profiling to occur, then information profiling would occur. And vice versa, if and only if that condition is not met, then no information profiling will occur. In addition, the count of number of times a conditional control transfer of a branch institution was taken illustrates that there are time-independent selection criteria of events.

5. As to claim 2, Chernoff teaches the computer of claim 1:

- the profile circuitry being configured to record at least one physical memory reference noting the source and destination of a control flow event in which control flow of the program execution diverges from sequential execution *("main memory", col. 1, lines 9-10, "in memory or within the registers of the new architecture, a set of bits which would be used to emulate the X86 condition codes", col. 3, lines 22-29, "FIG. 22 is a block diagram of a pair of data structures stored in memory which represents a return address stack", col. 4, lines 62-65, "converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system", see Abstract).*

Chernoff teaches the divergence from sequential execution when it converts instructions from a non native computer system to a native computer system.

6. As to claim 3, Chernoff teaches the computer of claim 1, wherein:

- the instruction pipeline is configured to execute instructions of an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction *("binary translator", Fig. 4, "pipeline instructions and thus execute the application program more quickly and efficiently", col. 20, lines 9-11, "pipelined fashion, executing multiple instructions in parallel", col. 19, lines 45-46. Chernoff teaches having a binary translator between two different modes, one with binary code and one without)*;
- the profile circuitry is configured to record profile information describing at least all events occurring during the profiled execution interval of the two classes: a divergence of execution from sequential execution, and a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor-mode before the mode change instruction (*"a profile file data structure 60 used to store information, col. 13, lines 62-67, see Fig. 5, "fields 70 are used to keep track or maintain count of the targets of the control transfer instruction", " number of times a conditional control transfer of a branch instruction was taken", col. 14, lines 64-67 through col. 15, lines 1-3, "converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system", see Abstract)*.

Chernoff teaches the divergence from sequential execution when it converts instructions/opcodes from a non native computer system to a native computer system.

7. As to claims 4 and 5, Chernoff teaches the computer of claim 3, wherein:

- the profile circuitry is configured to detect the occurrence of profileable events occurring in the instruction pipeline *("a profile file data structure 60 used to store information, col. 13, lines 62-67, see Fig. 5, "fields 70 are used to keep track or maintain count of the targets of the control transfer instruction", " number of times a conditional control transfer of a branch instruction was taken", col. 14, lines 64-67 through col. 15, lines 1-3, "pipeline instructions and thus execute the application program more quickly and efficiently", col. 20, lines 9-11, "pipelined fashion, executing multiple instructions in parallel", col. 19, lines 45-46);*
- to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention *("pipeline instructions and thus execute the application program more quickly and efficiently", col. 20, lines 9-11, "pipelined fashion, executing multiple instructions in parallel", col. 19, lines 45-46, see Figure 1, "condition codes are maintained on an X86 machine to provide knowledge of the machine state", col. 2, lines 64-66, "query the condition code", "conditional branch", col. 3, lines 4-7, "storing, in response to an execution of a condition code modifying instruction, a first table of methods, the methods providing access to evaluation routines which, when executed, modify one or all of the plurality of condition code bits", col. 3, lines 50-61).*

Chernoff does not teach using software intervention and does not teach first storing the profile information into main memory.

8. As to claims 6 and 27, Chernoff teaches the computer of claim 3, wherein the profile circuitry is configured to record profile information efficiently tailored to identify all bytes of object code executed during the profiled interval, without reference to the binary code of the program (*"a profile file data structure 60 used to store information, col. 13, lines 62-67, see Fig. 5, "fields 70 are used to keep track or maintain count of the targets of the control transfer instruction", " number of times a conditional control transfer of a branch instruction was taken", col. 14, lines 64-67 through col. 15, lines 1-3, "converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system", see Abstract, "The interpreter 44, for each instruction, determines the length or number of bytes comprising the instruction, identifies the opcode portion of the instruction, and determines the resources needed by the instruction.", col. 12, lines 41-47, "The run-time system collects profile data in response to execution of the native instructions to determine execution characteristics of the non-native instruction. Thereafter, the non-native instructions and the profile statistics are fed to a binary translator", see Abstract).* Chernoff teaches the divergence from sequential execution when it converts instructions/opcodes from a non native computer system to a native computer system.

9. As to claim 10, Chernoff teaches the computer of claim 1, wherein the program is executed on a computer having:

- an instruction pipeline configured to execute instructions from a memory of the computer (*"pipeline instructions and thus execute the application program more quickly and*

efficiently", col. 20, lines 9-11, "pipelined fashion, executing multiple instructions in parallel", col. 19, lines 45-46);

- profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline *("a profile file data structure 60 used to store information, col. 13, lines 62-67, see Fig. 5, "fields 70 are used to keep track or maintain count of the targets of the control transfer instruction", " number of times a conditional control transfer of a branch instruction was taken", col. 14, lines 64-67 through col. 15, lines 1-3, "pipeline instructions and thus execute the application program more quickly and efficiently", col. 20, lines 9-11, "pipelined fashion, executing multiple instructions in parallel", col. 19, lines 45-46);*
- to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention (*"pipeline instructions and thus execute the application program more quickly and efficiently", col. 20, lines 9-11, "pipelined fashion, executing multiple instructions in parallel", col. 19, lines 45-46, see Figure 1, "condition codes are maintained on an X86 machine to provide knowledge of the machine state", col. 2, lines 64-66, "query the condition code", "conditional branch", col. 3, lines 4-7, "storing, in response to an execution of a condition code modifying instruction, a first table of methods, the methods providing access to evaluation routines which, when executed, modify one or all of the plurality of condition code bits", col. 3, lines 50-61).*

10. As to claim 11, it is rejected for the same reasons as stated in the rejection of claim 5.

11. As to claim 14, Chernoff teaches the computer of claim 1, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer, and at least one of the recorded instruction references records the event of a sequential execution flow across a page boundary in the address space *("first pointer", "second pointer", col. 3, lines 50-67 through col. 4, lines 1-8)*.

12. As to claim 17, Chernoff teaches the computer of claim 1, wherein the recorded profile information indicates ranges of instruction binary text executed by the computer during a profiled interval of the execution *(see the rejection of claim 1)*, but fails to explicitly teach the ranges of executed text being recorded as low and high boundaries of the respective ranges. However, it is inherent that addresses in memory, which store data in code, are finite and bounded within a range (high boundary and low boundary).

13. As to claim 19, Chernoff teaches the computer of claim 1, wherein:

- the captured profile information comprises subunits of two kinds, a first subunit kind describing an instruction interpretation mode at an instruction boundary *("converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system", see Abstract);*

- a second subunit kind describing a transition between processor modes *("converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system", see Abstract).*

14. As to claim 20, it is rejected for the same reasons as stated in the rejection of claim 3. Applicant uses terminology such as "initiating events" and "non-initiating events" to describe whether an event of this event code is allowed to initiate collection of a new profile packet, or whether this event class may only be recorded in entries after the first. Claim 20 is similar to claim 3 but with different defined terminology by the Applicant. In addition, it is inherent that when a triggering event is detected, a non-triggering event is ignored. Similarly, when a triggering event is detected, an event will be triggered. Furthermore, it is inherent that the criteria takes into account both initiating and non-initiating events because the criteria is what distinguishes the event from being initiating or non-initiating.

15. As to claim 22, Chernoff teaches the computer of claim 1, the profile circuitry being further designed to record an event code describing the class of each profileable event recorded *("condition codes", col. 2, lines 64-67).*

16. As to claim 24, Chernoff teaches the computer of claim 1:

- wherein the instruction pipeline is configured to execute instructions of two instruction sets:

- a native instruction set providing access to substantially all of the resources of the computer *("main memory", col. 1, lines 9-10, "in memory or within the registers of the new architecture, a set of bits which would be used to emulate the X86 condition codes", col. 3, lines 22-29, "FIG. 22 is a block diagram of a pair of data structures stored in memory which represents a return address stack", col. 4, lines 62-65, "converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system", see Abstract)*;
- a non-native instruction set providing access to a subset of the resources of the computer *("main memory", col. 1, lines 9-10, "in memory or within the registers of the new architecture, a set of bits which would be used to emulate the X86 condition codes", col. 3, lines 22-29, "FIG. 22 is a block diagram of a pair of data structures stored in memory which represents a return address stack", col. 4, lines 62-65, "converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system", see Abstract)*.

17. As to claim 25, it is rejected for the same reasons as stated in the rejection of claim 24.

18. As to claim 28, it is rejected for the same reasons as stated in the rejection of claim 1.

19. As to claim 29, it is rejected for the same reasons as stated in the rejection of claim 2.

20. As to claim 30, it is rejected for the same reasons as stated in the rejection of claim 3.

21. As to claims 31 and 34, it is rejected for the same reasons as stated in the rejection of claim 1.

22. As to claim 33, it is rejected for the same reasons as stated in the rejection of claims 3 and 4.

23. As to claim 35, it is rejected for the same reasons as stated in the rejection of claim 14.

24. As to claim 36, it is rejected for the same reasons as stated in the rejection of claim 17

25. As to claim 37, it is rejected for the same reasons as stated in the rejection of claim 24

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 7-8, 15-16, 18, and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chernoff et al. (hereinafter Chernoff) (US 6,000,028).

26. As to claim 7, Chernoff teaches the computer of claim 1, wherein the profile circuitry is configured to record profile information during the profiled execution interval (*"a profile file data structure 60 used to store information, col. 13, lines 62-67, see Fig. 5, "fields 70 are used to keep track or maintain count of the targets of the control transfer instruction", " number of times a conditional control transfer of a branch instruction was taken", col. 14, lines 64-67 through col. 15, lines 1-3)*, but fails to explicitly teach:

- identifying each distinct physical page of instruction text executed;

However, "Official Notice" is taken that both the concept and advantages of providing that each distinct physical page of instruction text executed by identified is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include identifying each distinct physical page of instruction text executed to the existing system for the reason of improving data processing by being able to identify instructions.

27. As to claim 8, it is rejected for the same reasons as stated in the rejection of claim 3. In addition, Chernoff teaches wherein the profile circuitry is configured to record profile information during the profiled execution interval (*"a profile file data structure 60 used to store information, col. 13, lines 62-67, see Fig. 5, "fields 70 are used to keep track or maintain count of the targets of the control transfer instruction", " number of times a conditional control transfer of a branch instruction was taken", col. 14, lines 64-67 through col. 15, lines 1-3)*, but fails to explicitly teach:

- identifying each distinct physical page of instruction text executed;

However, "Official Notice" is taken that both the concept and advantages of providing that each distinct physical page of instruction text executed by identified is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include identifying each distinct physical page of instruction text executed to the existing system for the reason of improving data processing by being able to identify instructions.

28. As to claim 15, "Official Notice" is taken that both the concept and advantages of providing that at least one of the recorded instruction references records the event of a page boundary of the address space occurring within a single instruction is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the data instructions point to addresses which store the event of a page boundary to the existing system for the reason of increasing the control of the program by having a method of data allocation.

29. As to claim 16, "Official Notice" is taken that both the concept and advantages of providing that stored data instruction references an address location sequentially adjacent in the logical address space is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include stroing data sequentially in the logical address space to the existing system for the reason of increasing the simplicity of data storage allocation. It is a common practice in software to iterate address locations with +1 or –1.

30. As to claim 18, Chernoff fails to explicitly teach wherein the profile circuitry is configured to record, as the high boundaries of the respective ranges, the address of the last byte of each respective range. However, "Official Notice" is taken that both the concept and advantages of providing that recording/storing can be performed at the high boundary of the range of memory is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include this feature of storing at the high boundary of the range of memory to the existing system for the reason of improving data allocation by utilizing all the storage capacity available.

31. As to claim 32, it is rejected for the same reasons as stated in the rejection of claim 7.

Claims 9, 21, 26, and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chernoff et al. (hereinafter Chernoff) (US 6,000,028) in view of Agrawal et al. (hereinafter Agrawal) (US 5,768,500).

32. As to claims 9, 26 and 38, Chernoff fails to explicitly teach wherein the triggering event is expiration of a timer. However, Agrawal discloses that timer interrupts can be used to interrupt the processor when a set threshold is exceeded *("a hardware timer interrupts the processor when waiting time at a particular address exceeds a preset threshold", col. 8, lines 14-19)*. It would have been obvious to one of ordinary skill in the art at the time the invention

was made to include the feature of a timer to trigger an event for the reason of improving the control of the system by having interrupts to change states.

33. As to claim 21, Chernoff fails to explicitly teach the profile circuitry being further designed to record a timestamp describing a time of the recorded events. However, Agrawal discloses the use of a timestamp data to track the time of events *("timestamp", col. 10, lines 31-43, col. 11, lines 59-61)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of timestamps for the reason of improving the organization of the system by being able to track the time of particular events, which is important data that can be utilized.

Claims 12-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chernoff et al. (hereinafter Chernoff) (US 6,000,028) in view of Hammond et al. (hereinafter Hammond) (US 5,638,525).

34. As to claim 12, Chernoff teaches a multiprocessor computer system *("synchronization in multiple CPU environments", col. 15, lines 57-59)* with an instruction pipeline, profile circuitry and trigger but fails to explicitly teach it all belonging to a first CPU of a multiprocessor. However, Hammond discloses using two different instruction sets which use two different processors and architectures *(CISC and RISC processors, col. 4, lines 50-53)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use separate processors to store information separately for the reason of increasing efficiency by

having similar tasks and data together with one processor, while utilizing the other processor for its resources when needed.

In addition, Chernoff teaches recording profile information in a multiprocessor computer system but fails to explicitly teach using a second CPU of the multiprocessor that is able to coordinate/communicate with a first CPU of the multiprocessor. However, "Official Notice" is taken that both the concept and advantages of providing that a multiprocessor/parallel computer system can coordinate with one processor to another is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include this feature of at least partially coordinating the operation of the first CPU based at least in part on the analysis of the collected profile data to the existing system for the reason of increasing efficiency by having the similar tasks and data together contained in one processor utilized with the other processor for its resources so they can be accessed or analyzed.

35. As to claim 13, Chernoff in view of Hammond teaches the use of a first CPU and a second CPU but fails to explicitly teach wherein the controlling is effected by altering the text of the program executed by the first CPU. However, "Official Notice" is taken that both the concept and advantages of providing that control be done by altering the text of a program is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of controlling effected by altering the text of the program executed by the first CPU to the existing system for the reason of increasing the control of the system. Having text would be how instructions would be relayed to the first CPU.

Allowable Subject Matter

36. Claim 23 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. The current prior art fails to explicitly teach wherein a number of bits used to record the event code is less than $\log_2$ of the number of distinguished event classes.

Response to Arguments

37. Applicant argues about claim 1(on page 12, 1st paragraph) that the invention is claiming things that happen in hardware (instruction pipeline) and Chernoff describes condition codes implemented entirely in software – Chernoff's hardware has no condition codes.

In response, Examiner respectfully disagrees. Condition codes are implemented with the use of memory, which is hardware *("A simplistic approach to emulating condition codes is to maintain, in memory or within the registers of the new architecture, a set of bits which would be used to emulate the X86 condition codes.", col. 3, lines 22-28).*

38. Applicant demands a reference or an affidavit for the reliance of each following "Official Notice" that was taken in the non-final rejection of paper #9.

a) identifying each distinct physical page of instruction text executed
b) at least one of the recorded instruction references records the events of a page boundary of the address space occurring within a single instruction
c) stored data instruction references an address location sequentially adjacent in the logical address space
d) recording high boundaries of the respective ranges, the address of the last byte of each respective range.

a) Ramaswamy et al. (US 6,453,292 B2) teaches distinct executable instruction texts being identified and recognized *("A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for identifying commands in recognized text, the method steps include inputting recognized text, processing the recognized text by augmenting words of the recognized text with a position relative to a hypothesized command boundary, determining feature functions in the processed recognized text in accordance with a set of feature functions, deciding whether the processed recognized text with feature functions identified includes a command, the decision being made base on weighting of feature functions and if a command is included, outputting the command.", col. 2, lines 8-20).*

b) Chrysos et al. (US 6,549,930 B1) teaches within a profiled event register that the 1-bit fields record events for the selected instruction and that this can be done in a single instruction *("A profiled event register 330 is partitioned into, for example, one bit fields. The 1-bit fields record events for the selected instruction. When an instruction is first selected, the register is cleared. Events could include cache misses, branch mispredicts, resource conflicts, traps and exception conditions, retire/abort/invalid, TLB misses, taken/non-taken, data dependency stall, resource dependency stalls, and so forth. Note, this implementation allows multiple events to be attributed to a single instruction. It should be noted that event information is collected for both retired and aborted instructions. In order to reduce the size of the event register 330, some of the bit fields can be used to record different types of mutually exclusive events depending on the opcode of the instruction.).*

c) Lin (US 5,507,028) teaches storing data instructions sequentially adjacent in the logical address space *("Instructions are executed in a computer according to a certain logical sequence.*

Each instruction resides in the memory at a specific address. Two successive instructions in the memory may be executed in sequential or non-sequential ordering. When two instructions are sequentially adjacent to each other the address of the second instruction is exactly the address of the first instruction incremented by the length of the first instruction code.", col. 4, lines 38-45).

d) Hoyt et al. (US 5,903,751) teaches recording the address of the last byte (high boundary) of each respective range *("A set-associative branch target buffer cache for predicting a next memory block to fetch, said branch target buffer cache comprising a plurality of branch instruction entries, each of said branch instruction entries storing information about an associated branch instruction stored in a memory, some of said branch instructions comprising more than one byte in sequential addresses wherein said branch instructions comprising more than one byte may cross over a memory block boundary, said plurality of branch instruction entries addressed in said branch target buffer cache by an address of a last byte of said associated branch instruction.", see claim 1).* In addition, Senter et al. (US 5,659,782) also teaches recording the address of the last byte (high boundary) of each respective range *("This extra comparison with the last byte of the address is required since the store could cross a quad-word page boundary or be unaligned.", col. 10, lines 54-56).* The last byte is stored before it is compared.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (703) 305-5334. The examiner can normally be reached on 8:30AM - 7:00PM, Monday through Thursday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 746-7140.

Kt
1/9/04

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100